iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IX **Month of publication:** September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46628

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IX Sep 2022- Available at www.ijraset.com

Intrusion Detection System of IPv6 Network

Mangalagowri B N[1], Dr. N. Sandeep Varma[2]
[1, 2]*Department of Information Science and Engineering, B M S College of Engineering,* Bangalore, India

Abstract: ***There are variations within the network of the differential computing system's underlying IPv6 network. There are different blind zones for security detection. It is thoroughly investigated how the automatic data processing system differs from the underlying IPv6 network.***
An enhanced underlying IPv6 network security detection module design approach is suggested for the multicomputer system based on this concept.
One gets the comprehensive module design plan. The new technologies like receiving thread domain, object forwarding domain, and file buffer are taken into account by the network security detection system. It can alleviate the challenges with security detection brought on by IPv6 network variations in the underlying computing system. Thus, IPv6 networks have a significantly lower number of blind zones.
Keywords: IPv6, network security detection, differential computer systems, and intrusion detection systems (IDS).

I. INTRODUCTION

With the proliferation of different computing systems, concerns about data network protection and other factors, such intrusion from the outside environment, etc., are becoming more and more important. It is now more important than ever to keep an eye on network vulnerabilities to ensure the stability of networks and computer systems.
A system for tracking network activity, keeping an eye on vulnerability logs, and conducting other types of information analysis on networks is network security monitoring. It is important to follow both the offensive and defensive steps to ensure device security. The network security monitoring system is a strong security defence because, under the monitoring system's premise, the computer equipment will be protected in real-time from internal and external intrusion, and as a result, the warning activation will be received if the threat risk happens. The specific differential network security issues can be successfully solved when the network security identification system and IPv6 system are combined. The issue of the nation's lack of traditional IPv4 addresses should be resolved by the IPv6 protocol. Therefore, the network made possible by IPv6 is made to support the current differential process.
A novel and enhanced IPv6 network security identification module architectural solution is provided for the multi-computing framework. For the modules, the thorough concept layout is obtained.

The following are recent improvements to the network vulnerability monitoring technique:
1) Receiving thread domain
2) Object forwarding domain and
3) File buffer are taken into account. It will deal with the problem of vulnerability discovery brought on by changes in the IPv6 network in the underlying operating system.

II. DESIGN OF SECURITY DETECTION MODULEOF IPV6 NETWORK

The architecture for receiving thread domain, object forwarding domain, and file buffer is being introduced, and the vulnerability tracking function for the differential operating framework supporting IPv6 network is being reviewed. The computer framework that supports the IPv6 paradigm of network protection detection evolves as a result.

A. Design of overall framework module

The underlying IPv6 network protection recognition software is specified using the differential machine approach. The programme consists of the traffic packet amplification module, the traffic gap problem-solving bottom fusion module, and the centre where the configuration of the data packet collection is set up as follows:
1) File buffer area
2) Receiving thread domain and
3) Forwarding connection object domain.

Full module structure diagram is shown in Figure 1

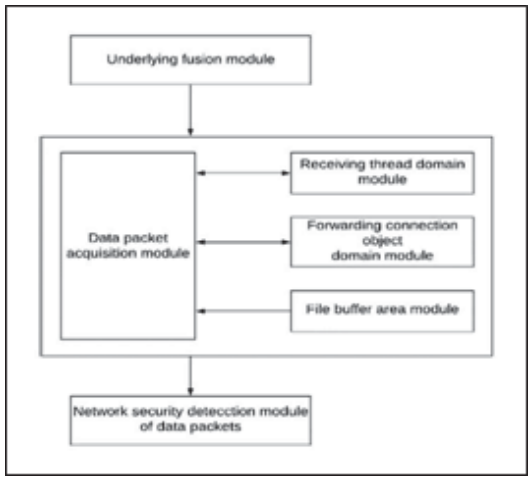


Fig.1. Full Module Structure

The above-described module architecture operates in accordance with the actions listed below :

- The data packets are first recorded in the Data Packet Acquisition Module, after which they go through the new technologies outlined above.
- The Network Security Detection Module of Data Packets detects any type of intrusion.

B. Data flow in a module

Flow map of the IPv6 network's underlying vulnerabilitydetection module designed as:

1) The framework for data collection is used to examine the data collected from the network link layer.

2) The redundant information contained in the data packet is eliminated.

We determine the information's consistency. When pertinent information is found, the linked thread is dropped from the thread domain, and the information is then examined. To transmit the data, the pertinent interaction artefacts are deleted from the forwarding entity domain. A new connection object should be created and the data transmitted to the recipient's address in cases where no communication entity is accessible. Before the file database is exhausted, the log information is kept in the database. The log information is subsequently added to the file. Figure 2 depicts the IPv6's network protection identification technique.

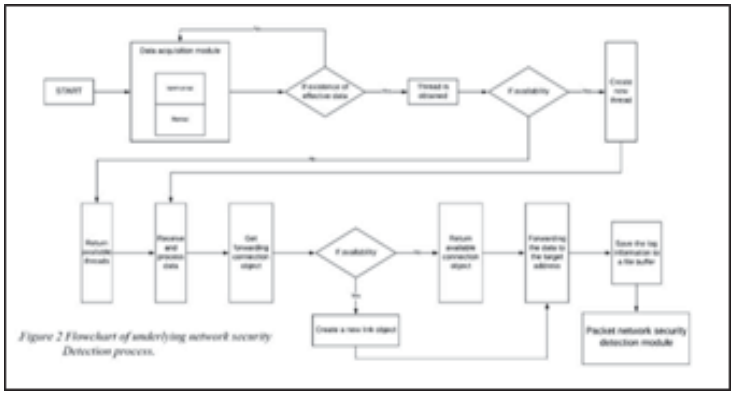


Fig. 2. The network protection identification mechanism

III. KEY TECHNOLOGY

A. Receiving Threads Domain Module Design

The receiving threads domain module, known as WinPcap, is a component of the vulnerability detection phase, and a BPF information filtering technique is needed to remove the useless data packet. Data packets' storage quality has increased. The data packet will be screened by WinPcap's virtual computer device driver module before being sent to the user module.

To improve the security control property of the entire device, WinPcap offers the proper Windows system data gathering technique that is compatible with Unix computer network administration capabilities. With the BPF kernel firewall, data forwarding is accomplished effectively. The data transfer request is sent by the data acquisition module utilising a split-thread domain when data acquisition is finished. The thread then resumes its previous state of rest. There is no requirement to use the Thread Domain method to create a new thread for each link. System output rises while computer energy consumption decreases.

B. Module Design Of Forwarding Connection Object Domain

The traditional forwarding module follows the object domain operating theory, which states that all previously used objects should be stored. The object can be programmed to use less energy if that is what is needed in order to utilise it again. When the machine develops connection items, certain resources are needed, and the object domain is used to reduce the total amount of device resources, it holds the relative artefacts. The following is a diagram of the link entity forwarding's operational steps: A position address is selected based on the set of routings after the data packets received from the thread have been retrieved and their location has been assessed. Fourth, the entity for the forwarding partnership is used to convey the data. Finally, the data packet is transmitted and the connection is severed once the forwarding relation entity establishes the connection to the destination address. Both the I/O technique and the termination route can be used to specify the destination address connection. It will spend a lot of money on such procedure. We build the link object domain, and if we decide to transfer data, a forwarding path object may be acquired for the data propagation. The link object returns the object domain once the data transmission is finished, and the enhanced method reduces the use of network sources while boosting the dependability of device data transfer.

C. Design Of File Buffer Module

The file buffer is intended to increase the effectiveness of data processing in the network. As the device performs I/O operations, it uses a lot of power. Data is written to the buffer area, and it is not written to the target machine until the buffer region has reached its capacity for data. and increase the I/O framework's effectiveness. Before it saturates the data, the data buffer stores the contents of the buffer region in the nearby system buffer state. The buffer system is then given access to all of the system's content. Resource utilisation is reduced as a result of recurrent read and write scripts and enhanced application efficiency. This method minimises resource usage by storing the log information in the log file after the data packet has been sent in the loop.

D. Design Of Packet Network Security Detection Module

Data transfer security, initial confirmation, and the prevention of recurrent attacks are all provided by the IPv6-based network security monitoring framework and other security technologies. The IPv6 protocol typically uses two protocols for knowledge protection to obtain the protected payload header and the header for authentication. The required protective action is carried out under the main command and control system. On the IP network, these programmes can typically be run independently. As the network stage and system layer carry out the attacking actions, the transfer of network data packets necessitates pause and trial. Data packets are handled and the network configuration is analysed layer by layer through the base network. High-rise device and application interface patterns are eventually noticed. To secure the stability of networks and computer systems, monitoring network vulnerabilities is becoming more and more important. The identification of network activity, vulnerability log monitoring, and other network information analysis are all part of the monitoring of network security. Vulnerability recognition is based on the database task's details of client functioning. The messages are sent to various analysis modules in accordance with the specific protocol form. If an assault takes place, this should warn. Finally, the differential operating approach makes it possible to find holes in the IPv6 network's foundation. With this system size, the bottom module eventually takes over as the application's primary communication node, and information security at the node's bottom has a significant impact on the entire network. The IPv6 network, the differential application structure, and the underlying vulnerability management method are all specifically discussed in this article. In addition, the necessary Detector Units are being developed.

E. Core Technology and Realization

The core of the differential computing device network protection identification framework, which is contained in the IPv6 network bottom module, is the network data packet processing element. Data packets related to security activities are collected in accordance with the relevant regulations, and then they are sent to the middle node for security evaluation. As a result, the entire security detection algorithm receives trustworthy data. To produce accurate findings for the whole network security detection framework, we must ensure the effectiveness of the network data collection software.

The main systems and the underlying network security detection differential computer system encounter the following challenges in accordance with the needs of the underlying IPv6 network safety detection software:

1) In the traditional section, the network packet would be lost if the data packet filter comprises a considerable number of details and we are unable to process the link quickly enough. The network data quality will be that, which will reduce the usefulness of upcoming research.
2) Problems with the device differential method's differential relations. efficient artefact protection via a machine differential approach.
3) Issue with different cache data files. After that, solutions for the aforementioned issues are provided, including the receiving thread scope, entity routing area, and file buffer.

IV. IMPLEMENTATION

A. Architecture

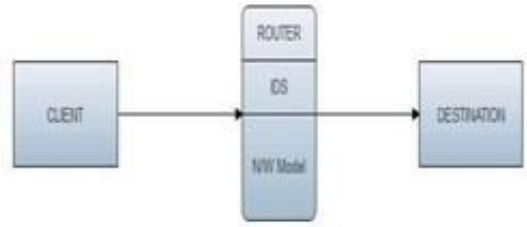


Fig. 3. Architecture

This is our project's fundamental architectural module, which is composed mostly of three blocks:

1) *Client*- A file will be created in this block, which will then divide it into smaller packets and transmit them to the router.
2) *Router*- After receiving packets from the client, it will create a forwarding table that will determine which router paths the packets should take. There is an IDS built into the router itself that can identify any attacks.
3) *Destination* – Once the destination block has successfully received all of the packets, message generation and reading can begin.

B. Use Case Diagrams

1) Client

a) The first client will haphazardly browse the file.
b) Produces random packets or gives the user the option to create intricate bespoke packets.
c) The packets will be transmitted using the first mode that is chosen.
d) In normal mode, the largest packet size is 48 bytes.
e) Attacker mode: 100 bytes maximum size per packet
f) After choosing a mode, the Router will receive packets that are sent in that manner.

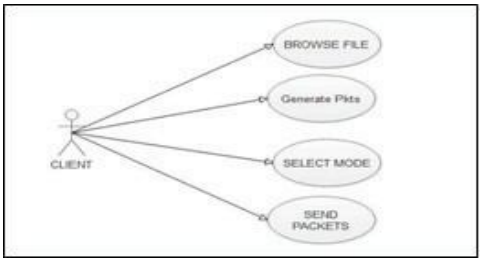


Fig. 4. Case Diagram – Client

2) Router

a) The router reads every packet that the client sends.
b) The IP address of the attacker will be kept for any future attacks when it checks for intruders.
c) Log buffer: The memory space used to store data prior to writing it to disk-based log files.
d) A packet will now be added to the transmission queue in order to reach its destination.
e) The packets will now begin to be received by the receiver.

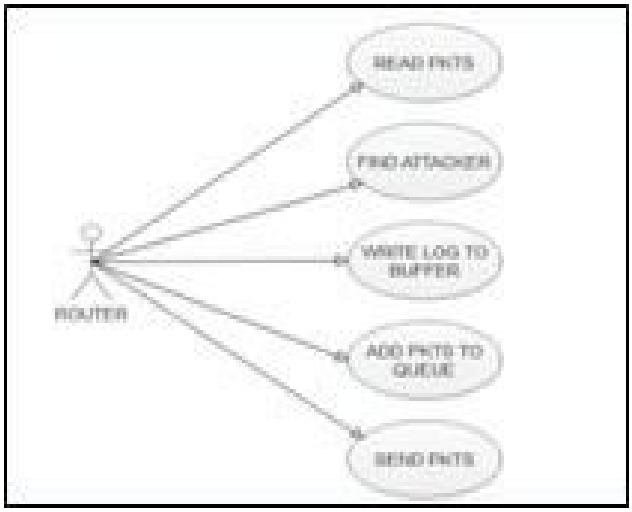


Fig. 5. Case Diagram – Router

3) Destination

The matching message will be generated when each packet has been read.

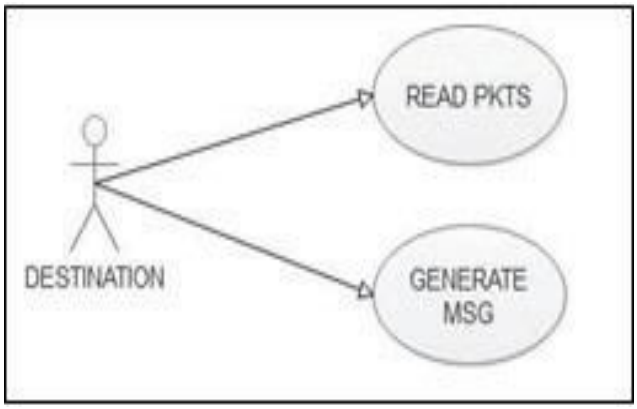


Fig. 6. Case Diagram- Destination

C. Proposed Block Diagrams

1) Level 0: This is the client-server system's outer model architecture. The client will initially produce the packets before beginning to transmit them to the router, where they will be forwarded to their destination in accordance with the router forwarding table.

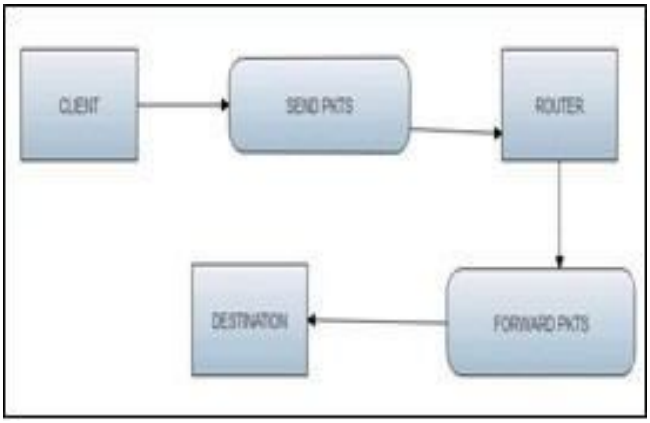


Fig. 7. Level 0

2) *Level 1:* This level displays the packet transmission between a client and a router. The user will first choose the modes (NORMAL/ATTACKER) that will determine how the packets are created (48 or 100 Bytes). After waiting and checking the state of the router queue, packets will be forwarded to the router if the queue is empty.

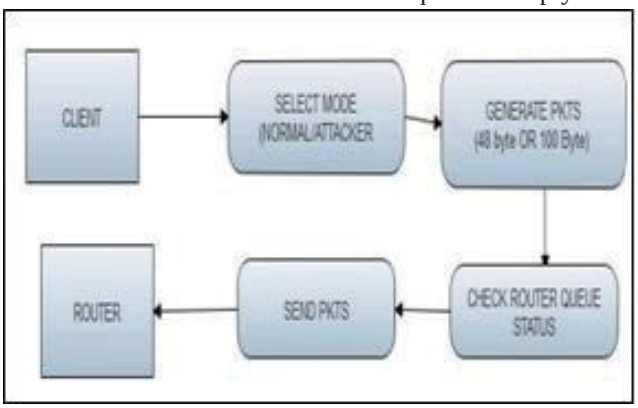


Fig. 8. Level 1

3) *Level 2:* The packet transport between the router and the target server is displayed at this level. Regardless of whether it is busy or idle, the first router reads the sender's request and responds. Then it reads the packets and determines the size of each one. If the packet is legitimate, it is forwarded to the destination; otherwise, if the packet is from an attacker, it is droppedand cached for later use.

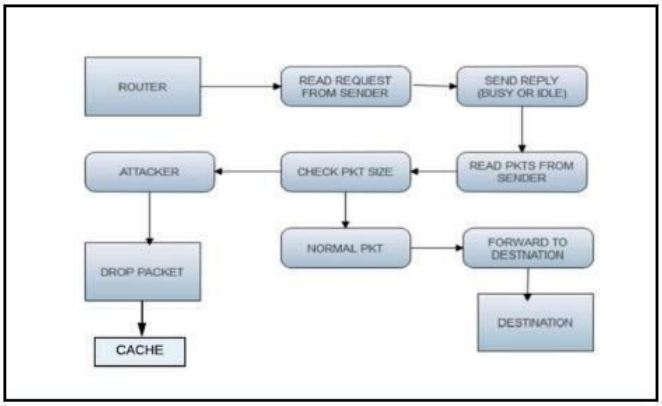


Fig. 9. Level 2

V. SIMULATIONS AND RESULTS

A. *Client Interface*

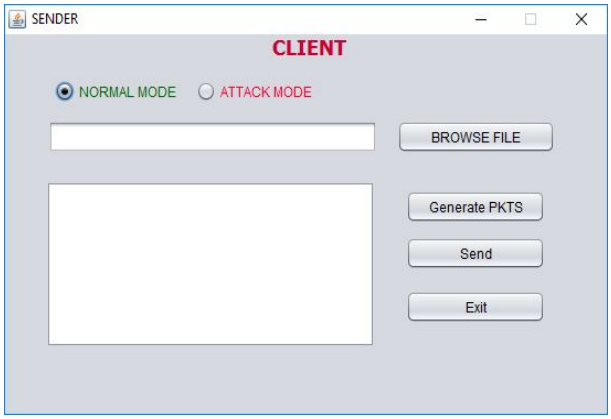


Fig. 10. Client Interface

Choose either the Normal mode or the Attack mode from the client interface's two modes. Browse the file that has to be transmitted after that, and then select Create Packets. When you eventually click "Send," the chosen file will be translated to the desired amount of packets based on the mode you've chosen.

B. IPv6 server interface

The number of packets in the queue and its size can always be seen in the IPv6 server interface. Additionally, the buffer size and limit can be shown. The overall status part displays the state of any intrusion, while the Attackers List section displays any IPs that have been detected as attacker IPs.

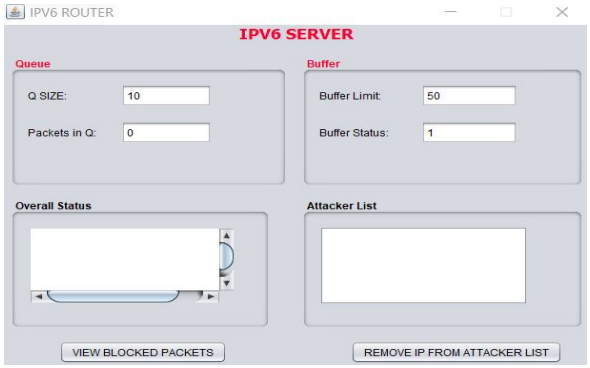


Fig. 11. IPv6 Server Interface

C. Receiver interface

Details like Sender IP, Total Packets, and Received Packet Number can be seen in the receiver interface. Additionally, the received message can be viewed under the Message area by selecting the View Message option.

Fig. 12. Receiver Interface

D. View blocked packets database

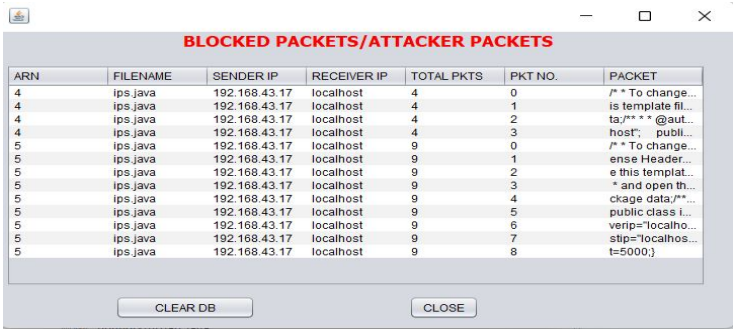


Fig. 13. View blocked packets database

In the Blocked Packets Database all the details of thepackets blocked are stored.

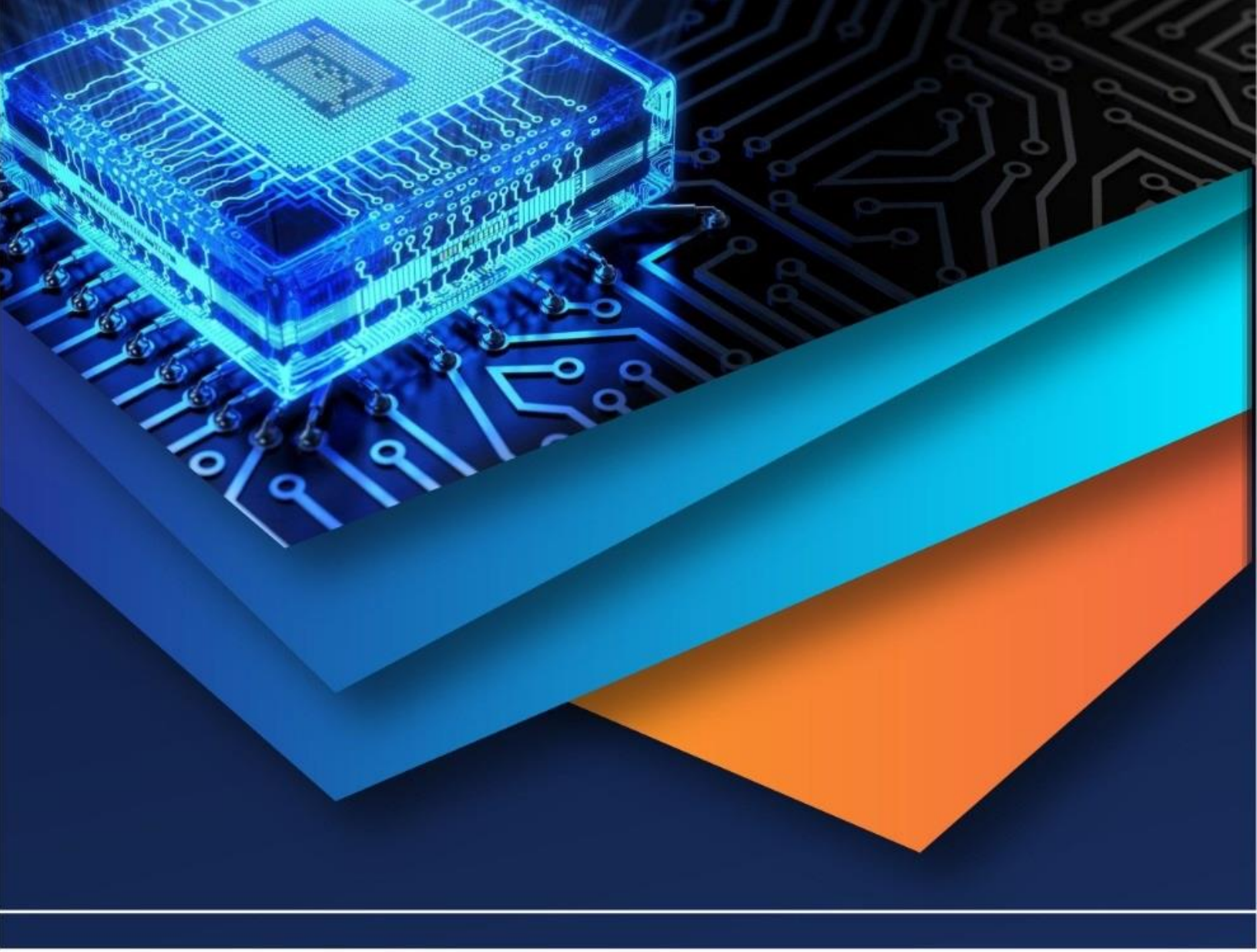
VI. CONCLUSION AND FUTURE WORK

We presented and discussed the Security Detection Module of the IPv6 Network. This takes into consideration recent advances like file buffers, object forwarding domains, and thread domain access. It has resolved the challenging issue of security identification brought on by differences in the underlying operating system's IPv6 network. We have proposed an enhanced underlying IPv6 network security identification module design procedure for the multicomputer architecture. Additionally, we set up an intrusion detection programme that can identify any incursion and block the IP address of the perpetrator. This can be used to thwart a variety of online threats. To locate the attacker inside the network, the header and packet sizes were limited. The IP address of the Attacker is Stored will be saved in a database and can be used to disable the Attacker if it poses a threat to the system in the future. The architecture displayed here offers fresh perspective on the stability and optimization of the IPv6 network. This is a novel point of view in comparison to the most recent technologies, and it has a number of issues that must be fixed for it to function well in the contemporary environment. Preliminary findings, however, indicate that performance can be fiercely competitive with that of present systems and offers the promise of doing so at a far lower level of processing complexity. Our findings should demonstrate how IPv6 networks have improved security and the ability to identify intrusions.

The model can be improved more in the future and additional factors can be taken into account. This model makes use of the security of data packets with IPv6 addresses. Due to the growing use of networking, hackers can stay one step ahead by exploiting translation processing techniques like NAT or tunnelling to change an IPv4 address to an IPv6 address and vice versa. The data and the network can be protected if the converted address can be identified before the data are compromised or if the defective packets are identified before entering the network.

REFERENCES

[1] Intrusion Detection System of IPV6 Based on Protocol Analysis by Zheng Zeng, 2016;
[2] MSIG. José Roberto Patio Sánchez, "Analysis of the Security IPv6 and Comparative Study between Two Routing Protocols Aware of IPv6", Asia-Pacific Conference on Computer Aided System Engineering, 2015;
[3] Security detection module of IPv6 networks, Fifth International Conference on Intelligent Systems Design and Engineering Applications, 2014; Zhuang Lian-Yingm1.
[4] "The BSD Packet Filter: A New Architecture for Userlevel Packet Capture," Steven McCanne and Van Jacobson San Diego: 2012 USEN Proceedings;
[5] "The BSD Packet Filter: A New Architecture for Userlevel Packet Capture," Steven McCanne and Van Jacobson San Diego: 2012 USEN Proceedings
[6] "FCFS SAVI: First-Come First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses," by E. Nordmark, M. Bagnulo, and E. Levy-Abegnoli. May 2012;
[7] M. Bagnulo and A. Garcia-Martnez, "SEND-based Source-Address Validation Implementation," September 2012.
[8] In September 2014, Rob Coltun, Dennis Ferguson, John Moy, and Acee Lindem published "OSPF for IPv6";
[9] In November 2014, Juan Luis Garcia Rambla published "Attacks on IPv4 and IPv6 Networks Data."
[10] "Analysis of IPv6 addressing and comparison analysis of routing protocols to IPv6," by Alexander Aguilar, Luis Chavez, and Jose Patio, published in November 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY